

**LISTING OF THE CLAIMS**  
**(including amendments, if any)**

1. **(currently amended)** A method for optimizing processing of a request, the request including one or more predicates, each predicate referencing one or more relations, the method including:
  - identifying the relations in the request;
  - identifying one or more access paths for each relation;
  - extracting identifying** the predicates **from in** the request;
  - for each predicate,
    - associating the predicate with the one or more access paths identified for the one or more relations referenced in the predicate;
    - estimating the costs of **executing the predicate using the** one or more access paths associated with the predicate; and
  - for **each at least one** access path **for which a cost was estimated**,
    - ~~selecting the cheaper of the estimated access path cost and an~~ **determining that a**
    - stored** actual access path cost **for the access path**, ~~if one exists, and, in~~
    - response, substituting the stored actual access path cost for the estimated**
    - access path cost, the stored actual access path cost for the access path**
    - being a measured cost of executing the predicate using the access path**;
  - for each predicate, for each relation referenced in the predicate, selecting an access**
  - path from among the one or more access paths associated with the predicate**;
  - ~~processing~~ **executing** the request using the selected access paths **for the predicates**,
  - measuring the actual cost of executing the predicates in the request; and**
  - ~~producing one or more actual path costs; and~~
  - storing the **measured actual costs as** ~~one or more~~ actual access path costs.
2. **(cancelled)**
3. **(currently amended)** The method of claim 1 where estimating the cost of **executing the predicates using the** one or more access paths includes:
  - retrieving estimated costs stored in a data dictionary.

4. **(currently amended)** The method of claim 1 where estimating the cost of executing the predicates using the one or more access paths includes:

performing selectivity costing based on one or more of selectivity, cardinality and statistics.

5. **(previously presented)** The method of claim 1 where selecting an access path further includes:

for each predicate,

selecting the lowest cost access path from among the access paths associated with each predicate.

6. **(previously presented)** The method of claim 1 where requests are sorted into workload groups and the method further includes:

categorizing the actual access path costs according to the workload group to which the request belongs.

7. **(previously presented)** The method of claim 6 where selecting an access path for the request taking into consideration a stored actual access path cost includes:

taking into consideration the categorized actual access path cost, if it exists, for the workload group to which the request belongs.

8. **(currently amended)** The method of claim 1 where storing the ~~one or more~~ measured actual ~~request element~~ costs includes:

storing the ~~one or more~~ measured actual ~~access path~~ costs in a cache.

9. **(currently amended)** The method of claim ~~[[1]]~~ 8 where storing the ~~one or more~~ measured actual ~~request element~~ costs includes:

backing up the stored ~~one or more~~ measured actual one or more measured actual ~~access path~~ costs from the cache to a query capture data base.

10. **(previously presented)** The method of claim 1 further including:

assigning the request to an one of a plurality of workload groups; and

where selecting an access path for the request includes taking into consideration a stored actual access path cost if the one of the plurality of workload groups is identified for such processing; and

where selecting an access path for the request does not include taking into consideration a stored actual access path cost if the one of the plurality of workload groups is not identified for such processing.

11. **(previously presented)** The method of claim 10 where storing the one or more actual request element costs includes:

performing a workload analysis to associate each actual access path cost with the workload group to which the request is assigned.

12. (original) The method of claim 11 where performing the workload analysis includes:

performing at least a portion of the workload analysis off-line.

13. (original) The method of claim 11 where performing the workload analysis includes:

performing at least a portion of the workload analysis in real time.

14. (original) The method of claim 1 further including:

creating one or more workload groups; and

assigning the request to one of the one or more workload groups.

15. **(currently amended)** A method for optimizing processing of a workload group, the workload group including one or more requests, at least one of the requests including one or more predicates, each predicate referencing one or more relations the method including:

identifying the relations in the at least one request request;

identifying one or more access paths for each relation;

~~extracting~~ identifying the predicates ~~from~~ in the request;

for each predicate,

associating the predicate with the one or more access paths identified for the one or more relations referenced in the predicate;

estimating the costs of executing the predicate using the one or more access paths associated with the predicate; and

for ~~each~~ at least one access path for which a cost was estimated,

~~selecting the cheaper of the estimated access path cost and an~~ determining that a  
stored actual access path cost for the access path, if one exists, and, in  
response, substituting the stored actual access path cost for the estimated  
access path cost, the stored actual access path cost for the access path  
being a measured cost of executing the predicate using the access path;

for each predicate, for each relation referenced in the predicate, selecting an access  
path from among the one or more access paths associated with the predicate;

~~processing~~ executing the request using the selected access paths for the predicates,

measuring the actual cost of executing the predicates in the request; and

~~producing one or more actual path costs; and~~

storing the measured actual costs as ~~one or more~~ actual access path costs, categorized by workload group.

16. (cancelled)

17. (currently amended) The method of claim 15 where estimating the cost of executing the predicates using the one or more access paths includes.

18. (currently amended) The method of claim 15 where estimating the cost of executing the predicates using the one or more access paths includes:

performing selectivity costing based on one or more of selectivity, cardinality and statistics.

19. (previously presented) The method of claim 15 where selecting an access path further includes:

for each predicate,

selecting the lowest cost access path from among the access paths associated with each predicate.

20. (previously presented) The method of claim 15 where the method further includes:

categorizing the actual access path costs according to the workload group to which the request belongs.

21. (previously presented) The method of claim 20 where selecting an access path for the request taking into consideration a stored actual access path cost includes:

taking into consideration the categorized actual access path cost, if it exists, for the workload group to which the request belongs.

22. (currently amended) The method of claim 15 where storing the ~~one or more~~ measured actual ~~request element~~ costs includes:

storing the ~~one or more~~ measured actual ~~access path~~ costs in a cache.

23. (currently amended) The method of claim ~~[[15]]~~ 22 where storing the ~~one or more~~ measured actual ~~access path~~ costs includes:

backing up the stored ~~one or more~~ measured actual ~~access path~~ costs from the cache to a query capture data base.

24. **(previously presented)** The method of claim 15 further including:

where selecting an access path for the request includes taking into consideration a stored actual access path cost if the workload group is identified for such processing; and  
where selecting an access path for the request does not include taking into consideration a stored actual access path cost if the workload group is not identified for such processing.

25. **(previously presented)** The method of claim 24 where storing the one or more actual request element costs includes:

performing a workload analysis to associate each actual access path cost with the workload group to which the request is assigned.

26. (original) The method of claim 25 where performing the workload analysis includes:

performing at least a portion of the workload analysis off-line.

27. (original) The method of claim 25 where performing the workload analysis includes:

performing at least a portion of the workload analysis in real time.

28. **(currently amended)** A computer program, stored on a tangible storage medium, for use in optimizing processing of a request, the request including one or more predicates, each predicate referencing one or more relations, the program including executable instructions that cause a computer to:

identify the relations in the request;

identify one or more access paths for each relation;

~~extract~~ identify the predicates ~~from in~~ the request;

for each predicate,

associate the predicate with the one or more access paths identified for the one or more relations referenced in the predicate;

estimate the costs of executing the predicate using the one or more access paths associated with the predicate; and

for ~~each~~ at least one access path for which a cost was estimated,

~~select the cheaper of the estimated access path cost and an~~ determine that a stored actual access path cost for the access path, if one exists, and, in response, substitute the stored actual access path cost for the estimated access path cost, the stored actual access path cost for the access path being a measured cost of executing the predicate using the access path;

~~process~~ execute the request using the selected access paths for the predicates,

measuring the actual cost of executing the predicates in the request; and

~~producing one or more actual path costs; and~~

store the measured actual costs as ~~one or more~~ actual access path costs.

29. **(cancelled)**

30. **(currently amended)** The computer program of claim 28 where when estimating the cost of executing the predicates using the one or more access paths the computer:

retrieves estimated costs stored in a data dictionary.

31. **(currently amended)** The computer program of claim 28 where when estimating the cost of executing the predicates using the one or more access paths the computer:

performs selectivity costing based on one or more of selectivity, cardinality and statistics.

32. **(previously presented)** The computer program of claim 28 where when selecting an access path the computer further:

for each predicate,

selects the lowest cost access path from among the access paths associated with each predicate.

33. **(previously presented)** The computer program of claim 28 where requests are sorted into workload groups and the computer program further includes executable instructions that cause the computer to:

categorize the actual access path costs according to the workload group to which the request belongs.

34. **(previously presented)** The computer program of claim 33 where when selecting an access path for the request taking into consideration a stored actual access path cost the computer:

takes into consideration the categorized actual access path cost, if it exists, for the workload group to which the request belongs.

35. **(currently amended)** The computer program of claim 28 where when storing the ~~one or more~~ measured actual ~~request element~~ costs the computer:

stores the ~~one or more~~ measured actual ~~access path~~ costs in a cache.

36. **(currently amended)** The computer program of claim ~~[[28]]~~ 35 where when storing the ~~one or more~~ measured actual ~~request element~~ costs the computer:

backs up the stored ~~one or more~~ measured actual ~~access path~~ costs from the cache to a query capture data base.

37. **(previously presented)** The computer program of claim 28 further including executable instructions that cause a computer to:

assign the request to an one of a plurality of workload groups;

where when selecting an access path for the request, the computer takes into consideration a stored actual access path cost if the one of the plurality of workload groups is identified for such processing; and



where when selecting an access path for the request, the computer does not include take into consideration a stored actual access path cost if the one of the plurality of workload groups is not identified for such processing.

38. **(previously presented)** The computer program of claim 37 where when storing the one or more actual access path costs the computer:

performs a workload analysis to associate each actual access path cost with the workload group to which the request is assigned.

39. (original) The computer program of claim 38 where when performing the workload analysis the computer:

performs at least a portion of the workload analysis off-line.

40. (original) The computer program of claim 38 where when performing the workload analysis the computer:

performs at least a portion of the workload analysis in real time.

41. (original) The computer program of claim 28 further including executable instructions that cause a computer to:

create one or more workload groups; and

assign the request to one of the one or more workload groups.

42. **(currently amended)** A database system including:
- a massively parallel processing system including:
    - one or more nodes;
    - a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs;
    - a plurality of data storage facilities each of the one or more CPUs providing access to one or more data storage facilities;
  - a process for execution on the massively parallel processing system for optimizing processing of a request, the request including one or more predicates, each predicate referencing one or more relations, the process including:
    - identifying the relations in the request;
    - identifying one or more access paths for each relation;
    - ~~extracting~~ identifying the predicates ~~from~~ in the request;
    - for each predicate,
      - associating the predicate with the one or more access paths identified for the one or more relations referenced in the predicate;
      - estimating the costs of executing the predicate using the one or more access paths associated with the predicate; and
    - for ~~each~~ at least one access path for which a cost was estimated,
      - ~~selecting the cheaper of the estimated access path cost and and~~ determining that a stored actual access path cost for the access path, if one exists, and, in response, substituting the stored actual access path cost for the estimated access path cost, the stored actual access path cost for the access path being a measured cost of executing the predicate using the access path;
    - ~~processing~~ executing the request using the selected access paths for the predicates, measuring the actual cost of executing the predicates in the request; and
    - ~~producing one or more actual path costs; and~~
    - storing the measured actual costs as one or more actual access path costs.

43. **(cancelled)**

44. **(currently amended)** The database system of claim 42 where estimating the cost of executing the predicates using the one or more access paths includes:

retrieving estimated costs stored in a data dictionary.

45. **(currently amended)** The database system of claim 42 where estimating the cost of executing the predicates using the one or more access paths includes:

performing selectivity costing based on one or more of selectivity, cardinality and statistics.

46. **(previously presented)** The database system of claim 42 where selecting an access path further includes:

for each predicate,

selecting the lowest cost access path from among the access paths associated with each predicate.

47. **(previously presented)** The database system of claim 42 where requests are sorted into workload groups and the process further includes:

categorizing the actual access path costs according to the workload group to which the request belongs.

48. **(previously presented)** The database system of claim 47 where selecting an access path for the request taking into consideration a stored actual access path cost includes:

taking into consideration the categorized actual access path cost, if it exists, for the workload group to which the request belongs.

49. **(currently amended)** The database system of claim 42 where storing the one or more measured actual ~~access-path~~ costs includes:

storing the one or more measured actual ~~access-path~~ costs in a cache.

50. **(currently amended)** The database system of claim 42 where storing the one or more measured actual ~~access-path~~ costs includes:

backing up the stored one or more measured actual ~~access-path~~ costs from the cache to a query capture data base.

51. **(currently amended)** The database system of claim 42 where the process further includes:

assigning the request to an one of a plurality of workload groups; and

where selecting an access path for the request includes taking into consideration a stored actual access path cost if the one of the plurality of workload groups is identified for such processing; and

where selecting an access path for the request does not include taking into consideration a stored actual access path cost if the one of the plurality of workload groups is not identified for such processing.

52. **(previously presented)** The database system of claim 51 where storing the one or more actual access path costs includes:

performing a workload analysis to associate each actual access path costs with the workload group to which the request is assigned.

53. (original) The database system of claim 52 where performing the workload analysis includes:

performing at least a portion of the workload analysis off-line.

54. (original) The database system of claim 52 where performing the workload analysis includes:

performing at least a portion of the workload analysis in real time.

55. (original) The database system of claim 42 where the process further includes:

creating one or more workload groups; and

assigning the request to one of the one or more workload groups.

56. (original) The database system of claim 42 further comprising a plurality of additional process for optimizing processing of a request.